

A Singularity-free Path Planner for Closed-chain Manipulators

Oriol Bohigas, Michael E. Henderson, Lluís Ros, and Josep M. Porta

Abstract—This paper provides an algorithm for computing singularity-free paths on non-redundant closed-chain manipulators. Given two non-singular configurations of the manipulator, the method attempts to connect them through a configuration space path that maintains a minimum clearance with respect to the singularity locus at all points. The method is resolution-complete, in the sense that it always returns a path if one exists at a given resolution, or returns “failure” otherwise. The path is computed by defining a new manifold that maintains a one-to-one correspondence with the singularity-free configuration space of the manipulator, and then using a higher-dimensional continuation technique to explore this manifold systematically from one configuration, until the second configuration is found. Examples are included that demonstrate the performance of the method on illustrative situations.

Index Terms—Closed-chain motion planning, singularity-free path, higher-dimensional continuation, singularity avoidance.

I. INTRODUCTION

A fundamental task in Robotics is the computation of collision-free motion paths between two configurations of a manipulator [1]. The subject has received substantial attention and, specially for open-chain manipulators, efficient algorithms exist that solve hard instances of the problem, even on cluttered environments and highly-dimensional spaces [2]. To compute a solution path, however, most of these algorithms require a parametric description of the configuration space (C-space), which complicates substantially their application to manipulators with closed kinematic chains. Such chains arise in a variety of contexts in Robotics, including parallel manipulators, multi-arm manipulation, or reconfigurable mechanisms, but they appear in other domains too, like on mechanical models of deployable structures, nano-devices, or biomolecules. In all of such contexts, a number of kinematic loop-closure constraints must be taken into account, which relate configuration parameters by non-linear equations, and thus induce a complex topological structure on the C-space. In general, as a result, this space may have several connected components and lower-dimensional singularity sets, and may not admit a global parameterization [3].

Despite the difficulty of the problem, some strategies have been given to synthesize feasible motion paths on closed-chain manipulators [4]–[10]. However, none of such works addresses the problem of avoiding the so-called *singular*

configurations, where the kinetostatic performance of the manipulator dramatically degrades. Although several types of singularities can be distinguished [11], those of primary interest in closed-chain manipulators are the *forward*, or *parallel* singularities, which compromise the velocity control of the manipulator and, dually, its capacity to withstand externally-applied forces, possibly leading to malfunction or breakage of the structure. In practice, thus, any path connecting two configurations must not only run through collision-free regions of the C-space, but also must avoid crossing the forward singularity locus at any point.

Several works in the Literature consider the problem of on-line singularity avoidance [12], but only a few of them tackle the more general problem of computing singularity-free paths between distant configurations. Solutions to this problem have been given for parallel manipulators only, and include an algorithm based on deforming a parametrized path between the query configurations [13], a variational approach that reduces the problem to a boundary value problem [14], and a numerical technique based on treating the singularity locus as a collection of obstacles [15]. All of these algorithms work well in favorable situations, but [13] and [14] mention limitations relative to proving path existence in certain cases, and the method in [15] is computationally intensive, as it requires constructing polytope approximations of the entire singularity set before searching for a feasible path. In some way or another, moreover, the methods in [13]–[15] exploit the fact that the considered C-spaces have closed-form parameterizations, and hence it is not trivial to extend them to tackle general kinematic structures.

The method provided in this paper, in contrast, makes no recourse to closed-form parameterizations, and hence can be applied to any non-redundant closed-chain manipulator. As opposed to [15], moreover, it treats singularities implicitly, not explicitly as obstacles, resulting in a computationally less intensive approach. The method relies on defining a system of equations whose solution manifold corresponds to the singularity-free subset of the C-space, so that maneuvering through such manifold guarantees singularity avoidance at all times. Then, an extension of the higher-dimensional continuation strategy given in [9] is defined, to explore this manifold systematically, until a path joining the start and goal configurations is found, or path non-existence is determined at a given resolution. The method, as presented, only avoids crossing singular configurations and neglects possible collisions of the manipulator. However, this is done for ease of explanation only, and incorporating the treatment of collisions only requires adopting the provisions made in [9] to this end.

O. Bohigas, L. Ros, and Josep M. Porta are with the Kinematics and Robot Design Group at the Institut de Robòtica i Informàtica Industrial, CSIC-UPC, Llorens Artigas 4-6, 08028 Barcelona, Spain. E-mails: {obohigas,lros,porta}@iri.upc.edu.

Michael E. Henderson is with the Numerical Analysis Group in the Mathematical Sciences Department at IBM's Thomas J. Watson Research Center, Yorktown Heights, NY USA. E-mail: mhender@us.ibm.com.

This work has been partially supported by the Spanish Ministry of Economy under contract DPI2010-18449.

The rest of the paper is organized as follows. Section II prepares the ground for the paper and describes the conditions that characterize the forward singularities of a manipulator, explaining their significance from a physical and geometrical point of view. Based on such background, Section III then describes the proposed singularity-free path planning method. Section IV shows the method's performance on illustrative situations. Section V, finally, provides the paper's conclusions and highlights several points deserving further attention.

II. PRELIMINARIES

The allowable positions and orientations of all links in a manipulator can always be encoded in a vector \mathbf{q} of n_q generalized coordinates, subject to a system of n_e equations

$$\Phi(\mathbf{q}) = \mathbf{0}, \quad (1)$$

which expresses the assembly constraints imposed by the joints [16]–[18]. Here, $\Phi(\mathbf{q}) : \mathcal{Q} \rightarrow \mathcal{E}$ is a differentiable map, and \mathcal{Q} and \mathcal{E} are n_q - and n_e -dimensional manifolds, respectively.

Let \mathcal{C} denote the C-space of the manipulator. That is,

$$\mathcal{C} = \{\mathbf{q} \in \mathcal{Q} : \Phi(\mathbf{q}) = \mathbf{0}\}. \quad (2)$$

In the usual setting, the differential $\Phi_{\mathbf{q}} = [\partial\Phi_i/\partial q_j]$ is full rank at all points $\mathbf{q} \in \mathcal{C}$, except on a zero-measure set of points \mathcal{G} where \mathcal{C} may lose the manifold structure. Thus, $\mathcal{C} \setminus \mathcal{G}$ can be thought to be a smooth manifold of dimension $d = n_q - n_e$. The points of \mathcal{G} are called *C-space singularities* [11] and typically occur in bifurcations, sharpnesses, or dimension changes (Fig. 1). Note that, despite $\Phi_{\mathbf{q}}$ is a differentiable function globally describing \mathcal{C} , all of such accidents may eventually occur in \mathcal{C} .

The vector \mathbf{q} will be assumed to contain a subvector \mathbf{v} of n_v coordinates, corresponding to the actuated degrees of freedom, or *inputs*, of the manipulator. This allows considering the partition $\mathbf{q} = [\mathbf{y}^T, \mathbf{v}^T]^T$, where \mathbf{y} encompasses the n_y coordinates of \mathbf{q} not present in \mathbf{v} , and to write (1) as

$$\Phi(\mathbf{y}, \mathbf{v}) = \mathbf{0}. \quad (3)$$

It will be further assumed that the manipulator is non-redundant, i.e., that $n_v = d$, which means that the number of inputs in \mathbf{v} is the lowest required to fix a configuration, and in particular that $n_y = n_q - n_v = n_q - d = n_e$.

To see the role of singular configurations, consider the time derivative of (3):

$$\Phi_{\mathbf{y}} \dot{\mathbf{y}} + \Phi_{\mathbf{v}} \dot{\mathbf{v}} = \mathbf{0}. \quad (4)$$

Note that for configurations \mathbf{q} on which $\Phi_{\mathbf{y}}$ is full rank we can write (4) in the equivalent form

$$\dot{\mathbf{y}} = -\Phi_{\mathbf{y}}^{-1} \Phi_{\mathbf{v}} \dot{\mathbf{v}}, \quad (5)$$

which provides the time rates of the \mathbf{y} coordinates in terms of the time rates of the input coordinates \mathbf{v} —i.e., the solution to the forward instantaneous kinematic problem of the manipulator. However, (5) only holds whenever the matrix $\Phi_{\mathbf{y}}$

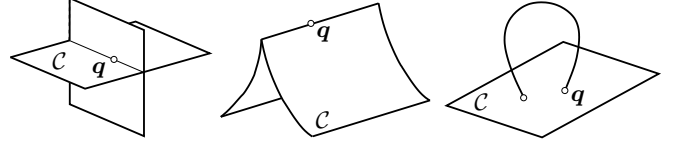


Fig. 1. Examples of C-space singularities.

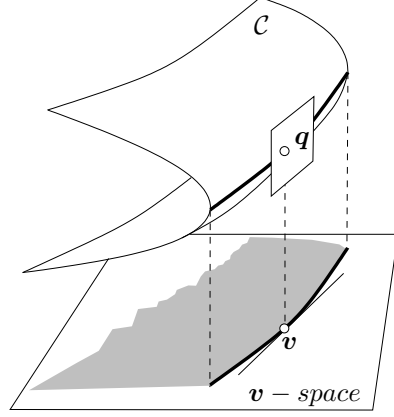


Fig. 2. Interpretation of points $\mathbf{q} \in \mathcal{C} \setminus \mathcal{G}$ where $\det(\Phi_{\mathbf{y}}) = 0$.

is full rank, and only in this case the input rates $\dot{\mathbf{v}}$ will determine a single value for the remaining rates $\dot{\mathbf{y}}$. This must be so because, if $\Phi_{\mathbf{y}}$ is rank-deficient at \mathbf{q} , then (4) yields, for a given value of $\dot{\mathbf{v}}$, either no solution or infinitely-many solutions for $\dot{\mathbf{y}}$, in which case it is not possible to uniquely determine the velocity of the manipulator by specifying the velocities of the actuators. Following these observations, a configuration $\mathbf{q} \in \mathcal{C}$ is said to be *singular* if $\det(\Phi_{\mathbf{y}}) = 0$.

Geometrically, the matrix $\Phi_{\mathbf{y}}$ may be rank deficient either because $\Phi_{\mathbf{q}}$ itself is rank deficient, or because only $\Phi_{\mathbf{y}}$ is. In the first case \mathbf{q} is in \mathcal{G} and in the second case the tangent space to \mathcal{C} at \mathbf{q} projects onto the \mathbf{v} -space as a linear variety of dimension lower than n_v (Fig. 2). In any case, both kinds of situations need to be avoided if a safe motion path for the manipulator is desired. A numerical method to obtain such paths is now given in the following section.

III. THE METHOD

Let \mathcal{C}_s denote the set of singular configurations, and let $\mathcal{C}_{sfree} = \mathcal{C} \setminus \mathcal{C}_s$. We call these sets the *singularity locus* and the *singularity-free C-space* of the manipulator, respectively. A singularity-free path joining two configurations of \mathcal{C}_{sfree} , \mathbf{q}_s and \mathbf{q}_g , is a continuous map $\tau : [0, 1] \rightarrow \mathcal{C}_{sfree}$, such that $\mathbf{q}(0) = \mathbf{q}_s$ and $\mathbf{q}(1) = \mathbf{q}_g$. The problem dealt with in this paper, thus, consists in computing such paths between two predefined configurations, \mathbf{q}_s and \mathbf{q}_g . This section proposes a new method to this end, based on defining a proper system of equations characterizing \mathcal{C}_{sfree} (Section III-A), and then resorting to a continuation method that uses these equations to trace \mathcal{C}_{sfree} systematically, until a path from \mathbf{q}_s to \mathbf{q}_g is found, or path non-existence is proved, at a considered resolution (Section III-B).

A. A system of equations defining $\mathcal{C}_{\text{free}}$

Observe that a configuration $\mathbf{q} = [\mathbf{y}^\top, \mathbf{v}^\top]^\top \in \mathcal{C}$ is non-singular if, and only if, it satisfies

$$\left. \begin{aligned} \Phi(\mathbf{q}) &= \mathbf{0} \\ \det(\Phi_{\mathbf{y}}) \cdot b &= 1 \end{aligned} \right\} \quad (6)$$

for some value of b , where $b \in \mathbb{R}$ is a newly-defined, auxiliary coordinate. Certainly, the first condition in (6) imposes \mathbf{q} to be a valid configuration, and the second condition holds at \mathbf{q} for some b whenever $\det(\Phi_{\mathbf{y}}) \neq 0$. In other words, (6) constitutes a system of equations that fully characterizes $\mathcal{C}_{\text{free}}$. For ease of explanation, (6) will be written as

$$\mathbf{F}(\mathbf{x}) = \mathbf{0} \quad (7)$$

hereafter, where

$$\mathbf{x} = [\mathbf{q}^\top, b]^\top, \quad (8)$$

and

$$\mathbf{F}(\mathbf{x}) = \begin{bmatrix} \Phi(\mathbf{q}) \\ \det(\Phi_{\mathbf{y}}) \cdot b - 1 \end{bmatrix}.$$

Let \mathcal{M} be the set of points \mathbf{x} that satisfy (7), and define the function $b(\mathbf{q}) = 1/\det(\Phi_{\mathbf{y}}(\mathbf{q}))$. Note that the points $\mathbf{x} \in \mathcal{M}$ are in one-to-one correspondence with the points $\mathbf{q} \in \mathcal{C}_{\text{free}}$, because for each $\mathbf{q} \in \mathcal{C}_{\text{free}}$ only the point $\mathbf{x} = [\mathbf{q}, b(\mathbf{q})]^\top$ satisfies (6). Accordingly, all paths on $\mathcal{C}_{\text{free}}$ are uniquely represented in \mathcal{M} , and viceversa. Thus, the original problem of computing a singularity-free path on \mathcal{C} from \mathbf{q}_s to \mathbf{q}_g can be reduced to that of connecting

$$\mathbf{x}_s = [\mathbf{q}_s^\top, b(\mathbf{q}_s)]^\top$$

and

$$\mathbf{x}_g = [\mathbf{q}_g^\top, b(\mathbf{q}_g)]^\top$$

through some path on \mathcal{M} . This reduction is advantageous because, by letting the path planner operate in \mathcal{M} , instead of in \mathcal{C} directly, guarantees that any computed path on \mathcal{M} will have a corresponding path on \mathcal{C} lying entirely in $\mathcal{C}_{\text{free}}$. This eliminates the need of checking singularity crossings in the planner, which, as noted in [13], may be difficult in some situations, due to the intricate structure of the singularity locus.

The correspondence of the two problems is schematically illustrated in Fig. 3. The horizontal plane at the bottom represents \mathcal{C} , which in this example coincides with the ambient space \mathcal{Q} for simplicity, and the singularity locus \mathcal{C}_s is represented by two red parabolas in this plane. To construct \mathcal{M} , we add a new dimension b to \mathcal{Q} (the vertical axis in the figure), and we lift every point $\mathbf{q} \in \mathcal{C}$ to the point $\mathbf{x} = [\mathbf{q}^\top, b(\mathbf{q})]^\top$. Then, \mathcal{M} can be thought of as a new manifold extending infinitely in the direction b , as the projection \mathbf{q} of a point $\mathbf{x} \in \mathcal{M}$ approaches \mathcal{C}_s .

Two important observations are in order regarding the search for a path. On the one hand, note that the differential $\mathbf{F}\mathbf{x}$ has the block structure

$$\mathbf{F}\mathbf{x} = \begin{bmatrix} \Phi_{\mathbf{q}} & \vdots & \mathbf{0} \\ \cdots & \cdots & \cdots \\ * & \vdots & \det(\Phi_{\mathbf{y}}) \end{bmatrix},$$

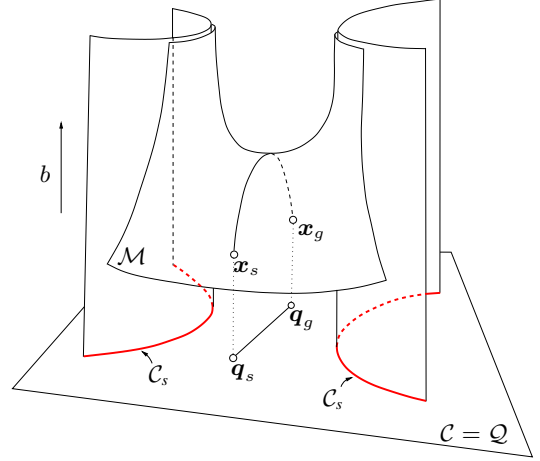


Fig. 3. The original problem of computing a singularity-free path on \mathcal{C} connecting \mathbf{q}_s and \mathbf{q}_g is transformed into one of finding an arbitrary path on \mathcal{M} connecting \mathbf{x}_s and \mathbf{x}_g .

from which we see that $\mathbf{F}\mathbf{x}$ is full rank at all points $\mathbf{x} \in \mathcal{M}$, because $\Phi_{\mathbf{y}}$ (and hence $\Phi_{\mathbf{q}}$) is full rank at such points. By the Implicit Function Theorem, this implies that \mathcal{M} has the structure of a smooth manifold everywhere [19], which is beneficial from the point of view of applying a continuation method to explore \mathcal{M} [20], because no bifurcations, sharpnesses, or dimension changes are expected to be found along the way, thus simplifying the numerical treatment of the problem. On the other hand, all of the \mathbf{q} coordinates have known bounds in practice [18], like those derived from mechanical limits on the joints, and $|b|$ should be maintained below a given threshold b_{\max} to guarantee some clearance from \mathcal{C}_s . As a result, the search for a path on \mathcal{M} must be restricted to a given domain \mathcal{D} of the \mathbf{x} -space, usually defined as the Cartesian product of a number of intervals derived from such bounds.

B. Exploring \mathcal{M} for a path

To determine a singularity-free path connecting \mathbf{x}_s and \mathbf{x}_g we can gradually construct an atlas of $\mathcal{M} \cap \mathcal{D}$. An atlas is a collection of charts where each chart C_i defines a local map between a domain $\mathcal{P}_i \subset \mathbb{R}^d$ and an open set around a given point $\mathbf{x}_i \in \mathcal{M}$, initially \mathbf{x}_s . The atlas will be computed using the higher-dimensional continuation method proposed in [20]. This method defines the local map for chart C_i using Ψ_i , an orthonormal basis of $T_{\mathbf{x}_i}\mathcal{M}$, the d -dimensional tangent space of \mathcal{M} at \mathbf{x}_i . The map is determined by first selecting a d -dimensional vector \mathbf{u}_j^i of parameters (Fig. 4-top), which is used to generate a point $\mathbf{x}_j^i \in \mathbb{R}^{n_q+1}$ in the neighborhood of \mathbf{x}_i as

$$\mathbf{x}_j^i = \mathbf{x}_i + \Psi_i \mathbf{u}_j^i. \quad (9)$$

Then, a point \mathbf{x}_j on \mathcal{M} is computed by orthogonally projecting \mathbf{x}_j^i . This projection is obtained by solving

$$\begin{aligned} \mathbf{F}(\mathbf{x}_j) &= \mathbf{0}, \\ \Psi_i^\top (\mathbf{x}_j - \mathbf{x}_j^i) &= \mathbf{0}, \end{aligned} \quad (10)$$

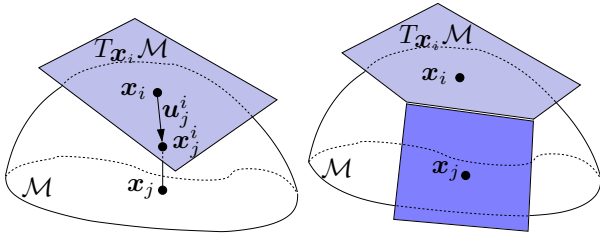


Fig. 4. The higher-dimensional continuation method applied to a two-dimensional manifold in 3D space. Left: A point x_j on \mathcal{M} can be obtained by orthogonally projecting a point x_j^i on $T_{x_i}\mathcal{M}$. Right: If a new chart is defined at x_j , it must be properly coordinated with the chart at x_i so that their projections smoothly covers the manifold.

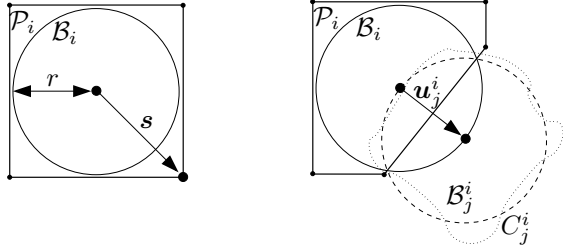


Fig. 5. Polytope-based chart construction. Left: The domain for chart C_i , \mathcal{P}_i , is a box including a ball of radius r around x_i . Right: \mathcal{P}_i is refined using a ball \mathcal{B}_j^i that approximates C_j^i , the projection on C_i of the part of the manifold covered by C_j .

using a Newton method initialized at x_j^i and where, at each iteration, x_j is updated with the increment Δx_j fulfilling

$$\begin{bmatrix} Fx_i \\ \Psi_i^T \end{bmatrix} \Delta x_j = - \begin{bmatrix} F(x_i) \\ \Psi_i^T (x_j - x_j^i) \end{bmatrix}. \quad (11)$$

The update is applied until the norm of the right-hand side of (11) becomes negligible or for a maximum number of iterations.

Each point on the manifold is the potential center of a new chart (see Fig. 4-bottom), and a method due to Henderson can be used to decide where to place the chart centers so as to ensure a good coverage of the manifold [20]. In his approach, the domain \mathcal{P}_i of chart C_i is initialized as a d -dimensional hypercube enclosing a ball \mathcal{B}_i of radius r , both defined in $T_{x_i}\mathcal{M}$, as illustrated in Fig. 5-top. A vertex of \mathcal{P}_i exterior to \mathcal{B}_i , with position vector s , is then used to generate a point x_j^i , using (9) with

$$u_j^i = \alpha \frac{s}{\|s\|}, \quad (12)$$

where α is initialized to r . If the projection from x_j^i to \mathcal{M} does not converge, or if the new chart C_j at x_j is too far or too different from C_i , i.e., if

$$\|x_j - x_j^i\| > \epsilon, \quad (13)$$

or

$$\|\Psi_i^T \Psi_j\| < 1 - \epsilon, \quad (14)$$

for a given threshold ϵ , the new chart is discarded and a new attempt of chart generation is performed with a smaller α .

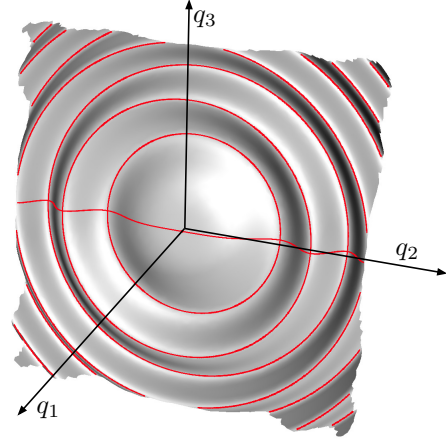


Fig. 6. A fictitious three-dimensional C-space with its singularities highlighted in red, assuming that q_1 and q_2 are the actuated degrees of freedom.

This procedure adapts the size of the area covered by each chart to the local curvature of the manifold. When C_j is valid, it is used to crop \mathcal{P}_i from the intersection between \mathcal{B}_i and C_j^i , the projection on $T_{x_i}\mathcal{M}$ of the part of the manifold covered by C_j . This projection is approximated by a ball \mathcal{B}_j^i of radius r in $T_{x_i}\mathcal{M}$, centered at the point given by u_j^i , as shown in Fig. 5-bottom. The intersection of \mathcal{B}_i and \mathcal{B}_j^i defines a new face of \mathcal{P}_i that eliminates some of its vertices (in particular the one given by s) and generates new ones. Symmetrically, the polytope \mathcal{P}_j associated with C_j is cropped using C_i . When C_i is surrounded by other charts, \mathcal{P}_i becomes a convex polytope included in \mathcal{B}_i , and C_i is considered to be *closed*, meaning that no further expansion of the atlas need to be attempted from that chart. Charts out of the domain \mathcal{D} are also considered closed. When all charts are closed, the connected component of \mathcal{M} containing the initial point x_s gets fully covered. If a singularity path exists from x_s to x_g , x_g must be covered by one of the charts of the atlas and, thus, a solution path can be determined by searching the graph implicitly defined by the chart centers and their neighborhood relations. If x_g is not covered by any of the charts of the atlas, path non-existence is established at the considered value for r .

The cost of the algorithm at each step is dominated by the cost of two searches among the set of charts: one to find the potential neighbors of a new chart, and another one to find an open chart from which to expand the atlas. The performance of the first search can be increased using a k -d tree storing the centers of the charts. To speed up the search for a solution path we resort to an A* search strategy [21]. In this strategy, the open charts are stored into a list sorted according to its minimum distance to x_s on the already explored part of the atlas plus a heuristic estimation of its distance to x_g . For the heuristic estimation we use the Euclidean distance, which is a lower bound of the actual distance on \mathcal{M} . Therefore, the heuristic is admissible and, consequently, the final path will be the shortest possible over \mathcal{M} , up to the resolution

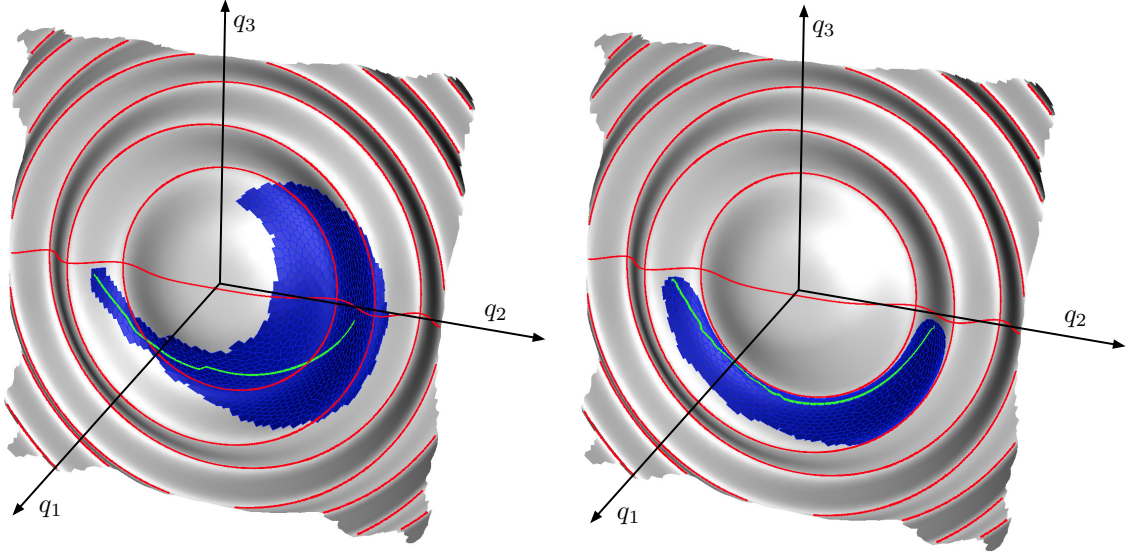


Fig. 7. The path computed by the proposed algorithm when neglecting and considering singularity avoidance (left and right, respectively). In the plots, the singularity locus is highlighted in red, the charts explored to connect the two query configurations are shown as blue polygons, and the final path is shown in green. While the path on the left figure crosses the singularity set twice, the path on the right figure is singularity-free.

used to define the atlas. If the sorted list of open charts is implemented using a heap, both the removal of the next chart to be expanded and the insertion of a new chart in the heap are logarithmic in the number of open charts.

IV. ILLUSTRATIVE EXAMPLES

The performance of the path planner is next illustrated on computing singularity-free paths in two situations: first on a fictitious three-dimensional C-space, and then on a 3RPR manipulator. The former case is chosen for its simplicity, to illustrate and visualize the method in detail, and the latter shows the method's performance on a real application. All results have been obtained on an implementation in C of the method, run on a iMac equipped with a 2.93 GHz Intel Core i7 processor.

A. A three-dimensional example

Consider the fictitious C-space defined implicitly by

$$\Phi(q_1, q_2, q_3) = q_1 - \sigma \cos(\omega (q_2^2 + q_3^2)) = 0, \quad (15)$$

with $\sigma = 0.5$ and $\omega = 0.25$. It is not difficult to see that this equation defines a sinusoidal surface in the space of $\mathbf{q} = [q_1, q_2, q_3]^T$, as shown in Fig. 6 for $\mathbf{q} \in [-1, 1] \times [-20, 20] \times [-20, 20]$.

Let us assume for this example that the vector of actuated degrees of freedom is $\mathbf{v} = [q_1, q_2]^T$, so that $\mathbf{y} = [q_3]$. Then, the singularities corresponding to such choice are given by the equation

$$\det(\Phi_{\mathbf{y}}) = \frac{\partial \Phi}{\partial q_3} = 2 \sigma \omega q_3 \sin(\omega (q_2^2 + q_3^2)) = 0, \quad (16)$$

which holds whenever $\omega (q_2^2 + q_3^2) = n \pi$ with $n \in \mathbb{Z}$, or when $q_3 = 0$. Thus the singularity locus is formed by a family of concentric circles and a sinusoidal line, which are

shown in red in Fig. 6. Note that, in accordance to Fig. 2, the points of such locus are those where the tangent plane to the C-space projects vertically on a line, on the space defined by $\mathbf{v} = [q_1, q_2]^T$.

Fig. 7 shows the results obtained by the planner, when trying to connect the configurations $\mathbf{q}_s = [0, 4.33, -0.38]^T$ and $\mathbf{q}_g = [0, -4.33, -0.38]^T$. To compare the results, the figure shows the computed path, in green, when singularities are not taken into account (left figure), and when their avoidance is considered (right figure). In both cases, the planner returns the shortest path up to the resolution of the generated atlas. The charts of this atlas are shown in blue in both figures, their shape becoming more clear when zooming the electronic version of the paper.

Note that the path on the left figure crosses the singularity locus twice, while the path on the right figure, although longer, avoids crossing any singularity. While the latter path approaches the singularity locus, we note that a certain clearance is always guaranteed, because the value of $|b|$ is always kept below a given threshold b_{max} . In this case, the value $b_{max} = 12$ was used, resulting in the shown path, but alternative paths with a larger clearance can be obtained if desired, by simply reducing b_{max} . In any case, nevertheless, b_{max} should always be chosen larger than the maximum of $|b(\mathbf{q}_s)|$ and $|b(\mathbf{q}_g)|$, so as to guarantee that the domain \mathcal{D} within which the search is restricted (Section III-A) includes \mathbf{x}_s and \mathbf{x}_g . The computation of the singularity-free path took 0.04 seconds in this example, using the continuation parameters $r = 0.25$ and $\epsilon = 0.25$.

B. A 3RPR manipulator

Now, let us consider the planar 3RPR manipulator in Fig. 8, consisting of a moving platform linked to the ground by means of three legs, where each leg is a revolute-

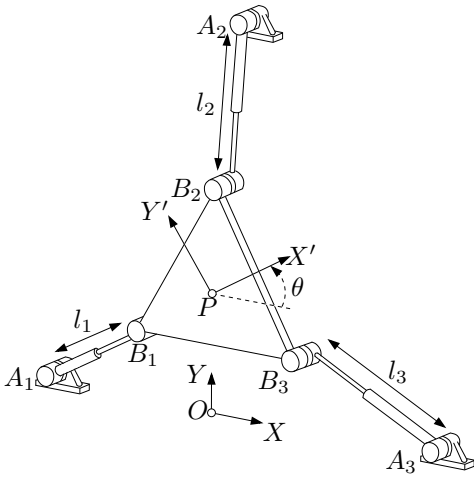


Fig. 8. A 3RPR parallel planar manipulator. Points A_1 , A_2 , and A_3 are fixed to the ground.

prismatic-revolute chain. The three prismatic joints are actuated, allowing to control the three degrees of freedom of the platform, and the remaining joints are passive.

It is well-known that, for a fixed value of the leg lengths l_1 , l_2 , and l_3 , this manipulator can be assembled into a maximum of six different configurations, also called *assembly modes*, and that certain pairs of such modes allow singularity-free transitions between them. Such transitions, which are desirable from the point of view of controllability [22], were thought to be impossible for a long time [23], but finally proved to exist by Innocenti and Parenti-Castelli [24]. Although they have been the object of further study [22, 25, 26], no previous work has shown how to compute them on general manipulators. In what follows, we show that our method is able to establish the particular transition found manually in [24].

To formulate (1), let \mathbf{a}_i and \mathbf{b}_i denote the position vectors of the anchor points A_i and B_i of the i th leg, relative to the fixed (OXY) and moving ($PX'Y'$) reference frames (Fig. 8). The leg lengths can be written in terms of the platform pose as

$$l_i^2 = \|\mathbf{p} + \mathbf{R}\mathbf{b}_i - \mathbf{a}_i\|^2, \quad i = 1, 2, 3 \quad (17)$$

where $\mathbf{p} = [x, y]^T$ is the position vector of P relative to the fixed frame, and \mathbf{R} is the 2×2 rotation matrix of angle θ . To replicate the same setting assumed in [24], we fix $l_1 = 14.98$ and assume the coordinates \mathbf{a}_i and \mathbf{b}_i given in Table I, so that (1) is the system formed by equations (17), with

$$\mathbf{q} = [x, y, \theta, l_2, l_3]^T. \quad (18)$$

This system implicitly defines a 2-dimensional C-space, because $d = n_q - n_e = 5 - 3 = 2$. In this case, moreover, since l_2 and l_3 are the actuated degrees of freedom, we have $\mathbf{v} = [l_2, l_3]^T$, and $\mathbf{y} = [x, y, \theta]^T$.

Fig. 9 shows the results obtained by the planner, when trying to connect the same two configurations considered

TABLE I
PARAMETERS OF THE CONSIDERED 3RPR MANIPULATOR.

| i | \mathbf{a}_i | \mathbf{b}_i | l_i |
|-----|----------------|----------------|---------|
| 1 | (0,0) | (0,0) | 14.98 |
| 2 | (15.91,0) | (17.04,0) | [10,40] |
| 3 | (0,10) | (13.33,16.10) | [10,40] |

in [24]. Namely,

$$\mathbf{q}_s = [14.674, -3.012, 2.132, 15.38, 12]^T, \quad (19)$$

$$\mathbf{q}_g = [-5.496, -13.935, -0.047, 15.38, 12]^T, \quad (20)$$

where the θ values are given in radians. As done in the previous example, this figure allows comparing the obtained paths when neglecting and considering singularity avoidance (left and right, respectively), using $b_{max} = 10^{-5}$. Such paths are shown overlaid onto the partial atlas generated (shown in blue), which in turn is plot onto the actual C-space (shown discretized as a triangular mesh). The singularity locus is a one-dimensional set in this case, drawn as a red curve in the figure. Note that for this manipulator we are only able to observe three-dimensional projections of the atlas and of the computed path, since the C-space is defined in a five-dimensional space \mathcal{Q} . Fig. 9 shows the particular projection obtained onto the coordinates l_2 , l_3 , and x . Note further how the computed path on the left figure crosses the singularity locus twice, while the path on the right figure maintains a large clearance with respect to such locus, because of the relatively low value of b_{max} employed. The computation of the singularity-free path took 0.35 seconds in this example, using $r = 0.75$ and $\epsilon = 0.25$.

V. CONCLUSIONS

This paper has introduced a novel approach to compute singularity-free paths on non-redundant closed-chain manipulators. Due to the complexity of the involved C-spaces, and of the associated singularity loci, previous attempts to solve this problem have only considered closed-form parametrizable C-spaces, like those arising in parallel manipulators. In contrast, the approach we present here makes no recourse to such parametrizations, and, thus, it can be applied to any non-redundant closed-chain manipulator in principle.

The problem has been tackled by defining a system of equations implicitly characterizing the singularity-free C-space of the manipulator, which avoids the need of representing the singularity locus explicitly as an obstacle. The solution manifold of this system, thus, can be freely navigated without fear of crossing any singularity of the manipulator. Higher-dimensional continuation techniques are then used to progressively construct an atlas of the component of this manifold that contains the start configuration, until the goal configuration is eventually reached, or path non-existence is proved, at the resolution of the atlas. When the query configurations cannot be connected, a complete atlas of the singularity-free component of the C-space that is reachable from the initial configuration is obtained. This atlas readily allows solving other planning queries within such component if desired.

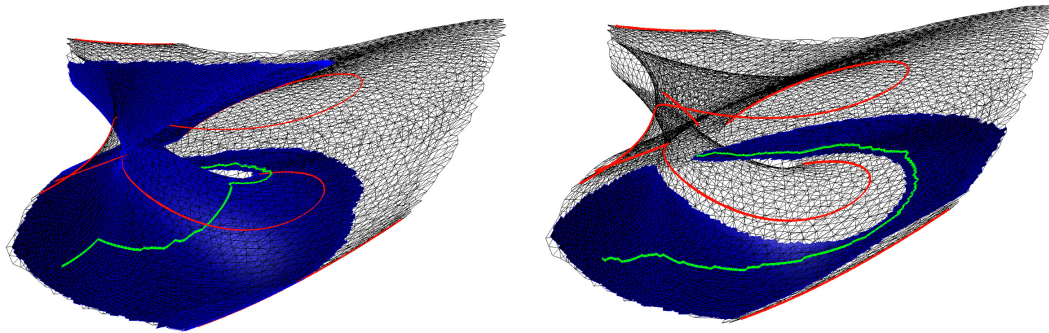


Fig. 9. The path computed by the proposed method on the test case described in Section IV-B. The figures show the path obtained (in green) when neglecting and considering singularity avoidance (left and right, respectively), together with the singularity curves (in red), the generated atlas (in blue) and the underlying C-space (approximated by a triangular mesh). The two plots correspond to a projection on the l_2 , l_3 , and x coordinates.

The focus of the paper has been on avoiding forward singularities, because they are those potentially harmful to the manipulator. In addition to forward singularities, however, closed-chain manipulators also exhibit inverse singularities, which give rise to dexterity losses of the end-effector. It is worth noting that the avoidance of such singularities might also be considered in our approach, since inverse singularities can be characterized in a way analogous to forward singularities, as certain configurations for which another Jacobian matrix is rank-deficient. For the sake of conciseness, moreover, we omitted the treatment of manipulator collisions throughout the paper, but they can be incorporated to the planner using the approach described in [9]. Finally, note that the resolution completeness of the presented approach comes at the expense of a computational cost that scales exponentially with the dimension of the C-space. To deal with higher-dimensional problems, however, we could adopt the approach in [10], which trades off resolution completeness by efficiency and probabilistic completeness. The evaluation of these two variants of the planner in the context of singularity-free path planning will be part of our future work.

REFERENCES

- [1] B. Siciliano and O. Khatib, Eds., *Springer Handbook of Robotics*. Berlin, Heidelberg: Springer, 2008.
- [2] S. LaValle, *Planning algorithms*. Cambridge Univ Press, 2006.
- [3] J. Burdick, “Kinematics and design of redundant robot manipulators,” Ph.D. dissertation, Stanford University, July 1988.
- [4] L. Han and N. M. Amato, “A kinematics-based probabilistic roadmap method for closed chain systems,” *Proceedings of the International Workshop on the Algorithmic Foundations of Robotics (WAFR)*, pp. 233–246, 2000.
- [5] J. H. Yakey, S. M. LaValle, and L. E. Kavraki, “Randomized path planning for linkages with closed kinematic chains,” *IEEE Transactions on Robotics*, vol. 17, no. 6, pp. 951–958, 2001.
- [6] J. Cortés and T. Simeon, “Probabilistic motion planning for parallel mechanisms,” in *Proceedings of the IEEE International Conference on Robotics and Automation*, vol. 3. IEEE, 2003, pp. 4354–4359.
- [7] J. Cortés and T. Siméon, “Sampling-based motion planning under kinematic loop-closure constraints,” *Algorithmic Foundations of Robotics VI*, pp. 75–90, 2005.
- [8] J. M. Porta, J. Cortés, L. Ros, and F. Thomas, “A space decomposition method for path planning of loop linkages,” *Proceedings of IEEE International Conference on Intelligent Robots and Systems*, pp. 1882–1888, 2007.
- [9] J. M. Porta and L. Jaillet, “Path planning on manifolds using randomized higher-dimensional continuation,” in *Algorithmic Foundations of Robotics IX*, ser. Springer Tracts in Advanced Robotics, D. Hsu, V. Isler, J.-C. Latombe, and M. Lin, Eds. Springer Berlin / Heidelberg, 2011, vol. 68, pp. 337–353.
- [10] L. Jaillet and J. M. Porta, “Path planning with loop closure constraints using an atlas-based RRT,” in *International Symposium on Robotics Research*, Flagstaff, USA, 2011.
- [11] D. Zlatanov, “Generalized singularity analysis of mechanisms,” Ph.D. dissertation, University of Toronto, 1998.
- [12] S. Bhattacharya, H. Hatwal, and A. Ghosh, “Comparison of an exact and an approximate method of singularity avoidance in platform type parallel manipulators,” *Mechanism and Machine Theory*, vol. 33, no. 7, pp. 965–974, 1998.
- [13] B. Dasgupta and T. Mruthyunjaya, “Singularity-free path planning for the Stewart platform manipulator,” *Mechanism and Machine Theory*, vol. 33, no. 6, pp. 711–725, 1998.
- [14] S. Sen, B. Dasgupta, and A. K. Mallik, “Variational approach for singularity-free path-planning of parallel manipulators,” *Mechanism and Machine Theory*, vol. 38, no. 11, pp. 1165–1183, 2003.
- [15] A. K. Dash, I.-M. Chen, S. H. Yeo, and G. Yang, “Workspace generation and planning singularity-free path for parallel manipulators,” *Mechanism and Machine Theory*, vol. 40, no. 7, pp. 776–805, 2005.
- [16] J. G. De Jalón and E. Bayo, *Kinematic and Dynamic Simulation of Multibody Systems*. Springer Verlag, 1993.
- [17] J. M. Porta, L. Ros, T. Creemers, and F. Thomas, “Box approximations of planar linkage configuration spaces,” *ASME Journal of Mechanical Design*, vol. 129, p. 397, 2007.
- [18] J. M. Porta, L. Ros, and F. Thomas, “A linear relaxation technique for the position analysis of multi-loop linkages,” *IEEE Transactions on Robotics*, vol. 25, no. 2, pp. 225–239, 2009.
- [19] S. G. Krantz and H. R. Parks, *The Implicit Function Theorem: History, Theory and Applications*. Boston: Birkhäuser, 2002.
- [20] M. E. Henderson, “Multiple parameter continuation: Computing implicitly defined k-manifolds,” *International Journal of Bifurcation and Chaos*, vol. 12, no. 3, pp. 451–476, 2002.
- [21] S. J. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*. Prentice Hall, 2003.
- [22] H. Bamberger, A. Wolf, and M. Shoham, “Assembly mode changing in parallel mechanisms,” *IEEE Transactions on Robotics*, vol. 24, no. 4, pp. 765–772, 2008.
- [23] K. H. Hunt and E. J. F. Primrose, “Assembly configurations of some in-parallel-actuated manipulators,” *Mechanism and Machine Theory*, vol. 28, no. 1, pp. 31–42, 1993.
- [24] C. Innocenti and V. Parenti-Castelli, “Singularity-free evolution from one configuration to another in serial and fully-parallel manipulators,” *Journal of Mechanical Design*, vol. 120, p. 73, 1998.
- [25] M. Zein, P. Wenger, and D. Chablat, “Non-singular assembly-mode changing motions for 3-RPR parallel manipulators,” *Mechanism and Machine Theory*, vol. 43, no. 4, pp. 480–490, 2008.
- [26] E. Macho, O. Altuzarra, C. Pinto, and A. Hernandez, “Transitions between multiple solutions of the direct kinematic problem,” *Advances in Robot Kinematics: Analysis and Design*, pp. 301–310, 2008.